UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/716,840	11/18/2003	Matthew Marcus	07844-619001	7089

21876 7590 11/20/2007

FISH & RICHARDSON P.C.
P.O. Box 1022
MINNEAPOLIS, MN 55440-1022

EXAMINER	
SAEED, USMAAN	
ART UNIT	**PAPER NUMBER**
2166	

MAIL DATE	DELIVERY MODE
11/20/2007	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

BEFORE THE BOARD OF PATENT APPEALS AND INTERFERENCES

Application Number: 10/716,840
Filing Date: November 18, 2003
Appellant(s): MARCUS, MATTHEW

Daniel J. Burns
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 8/22/2007 appealing from the Office action mailed 03/08/2007.

(1) Real Party in Interest

A statement identifying by name the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

The amendment after final rejection filed on 7/30/2007 has been entered.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

2002/0156772	Chau	10-2002

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claim Rejections - 35 USC § 102

The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

Claims 1- 34 and 37-40 are rejected under 35 U.S.C. 102(b) as being anticipated by **Chau et al.** (**Chau** hereinafter) (U.S. PGPub No. 2002/0156772).

With respect to claim 1, **Chau** teaches **a method for searching for one or more logical elements in a hierarchical tree structure of an extensible markup language XML document conforming to a schema used for XML, comprising:**

"providing a representation of an extensible markup language XML document instance containing two or more logical elements, wherein at least one

logical element is a parent node and at least one logical element is a child node in a hierarchical tree structure describing the representation" as XML Path or XPath addresses parts of an XML document. XPath gets its name from its use of a path notation as in URLs for navigating through the hierarchical structure of an XML document (**Chau** Paragraph 0042). Represents the element with the name tagn, which is a child of the descending chain from root, tag1, tag2, . . . where tag3 has the value "Los Angeles" (**Chau** Paragraph 0112).

"receiving a query for logical elements satisfying an Xpath expression" as the XML System also allows overrides of query conditions explicitly or implicitly defined in the DAD, by parsing the SQL or XML XPath based override parameter to the composition stored procedures (**Chau** Paragraph 0080).

"searching in the hierarchical tree structure only nodes that potentially have child nodes satisfying the Xpath expression" as the side tables are created by the DAD, and indices are created for columns in the side tables. Therefore, the search will be fast with indexing. Note that the invoice_number is the primary key in the application table sales_tab. The advantage of direct query with sub-query is better performance. When side tables have parent-children relationships, direct query with sub-query often make more sense (**Chau** Paragraph 0335 & 0336). Represents the element with the name tagn, which is a child of the descending chain from root, tag1, tag2, . . . where tag3 has the value "Los Angeles" (**Chau** Paragraph 0112). The FROM clause defines the tables containing the data, and the WHERE clause specifies the join

and search conditions (**Chau** Paragraph 0679). Examiner interprets that every node in this reference has a potential child nodes.

"providing the logical elements satisfying the XPath expression" as XML System provides extracting UDFs to retrieve XML elements or attributes in the SQL select clause. This is very useful after search filtering on a collection of XML documents to further obtain desired elements or attributes (**Chau** Paragraph 0318).

Claim 20 is essentially the same as claim 1 except that it sets forth the claimed invention as a computer program product and is rejected for the same reason as applied hereinabove.

With respect to claims 2 & 3, **Chau** teaches **"the method of claim 1, including the further step of generating a collection of parent nodes that potentially have child nodes satisfying the Xpath expression from a table relating a class of parent nodes/parent nodes and a class of child nodes/child nodes, and wherein the table is used in the final searching step"** as for the root element_node, all tables storing its attribute or all child element data should be specified (**Chau** Paragraph 0164). Representing an XML element. It must be defined in the specified DTD. For the RDB_node mapping, the root element_node must have a RDB_node to specify all tables containing XML data for itself and all its children nodes. It can have zero or more

attribute_nodes and child element_nodes, as well as zero or one text_node (**Chau** Paragraph 0176).

Claims 21 & 22 are essentially the same as claims 2 & 3 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claims 4 & 5, **Chau** teaches **"the method of claim 2, wherein the table comprises entries containing hash representations of a class of parent nodes/parent nodes and a class of child nodes/child nodes"** as levelmap is an associative array that maps column names to their equivalence class numbers or "relational level." The equivalence classes in ascending order of relational levels should have one-to-many relationship between each adjacent classes with the "many" side at the upper level. In the example, "order_key" maps to 0; "part_key" maps to 1; and "date" maps to 2. The associative array can be implemented in memory, for example, as a hash table, a sorted array, or a binary search tree (**Chau** Paragraph 0699). Create a hash table or stored array of all top_elements rel->top_elements for fast search (**Chau** Paragraph 0941).

Claims 23 & 24 are essentially the same as claims 4 & 5 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 6 & 7, **Chau** teaches **"the method of claim 2, wherein the table comprises a listing of permitted classes of child nodes/child nodes for each class of parent node/parent node"** as for the root element_node, all tables storing its attribute or all child element data should be specified (**Chau** Paragraph 0164). Representing an XML element. It must be defined in the specified DTD. For the RDB_node mapping, the root element_node must have a RDB_node to specify all tables containing XML data for itself and all its children nodes. It can have zero or more attribute_nodes and child element_nodes, as well as zero or one text_node (**Chau** Paragraph 0176 & 0119). Tables contain lists of data, therefore this reference contain table with lists of root/parent nodes and children nodes. Examiner refers these non restricted paths as permitted paths.

Claims 25 & 26 are essentially the same as claims 6 & 7 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 8 & 9, **Chau** teaches **"the method of claim 2, wherein the table comprises a listing of permitted classes of parent nodes/parent nodes for each class of child node/child node"** as for the root element_node, all tables storing its attribute or all child element data should be specified (**Chau** Paragraph 0164). Representing an XML element. It must be defined in the specified DTD. For the

RDB_node mapping, the root element_node must have a RDB_node to specify all tables containing XML data for itself and all its children nodes. It can have zero or more attribute_nodes and child element_nodes, as well as zero or one text_node (**Chau** Paragraph 0176 & 0119). Tables contain lists of data, therefore this reference contain table with lists of root/parent nodes and children nodes. Examiner refers these non restricted paths as permitted paths.

Claims 27, and 28 are essentially the same as claims 8 & 9 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 10 & 11, **Chau** teaches **"the method of claim 1, further comprising:**

"receiving a rule set identifying allowable combinations between child nodes and parent nodes in a hierarchical document structure" as Extensible Markup Language (XML) is a set of rules or guidelines for designing text formats for structured data using tags (**Chau** Paragraph 0040). Represents the element with the name tagn which is a child of the descending chain from root, tag1, tag2, . . . where tag3 has the value "Los Angeles" (**Chau** Paragraph 0112 & 0119). Examiner refers these non-restricted paths as permitted paths and these permitted paths contain allowable combinations.

"transforming the rule set into a table relating a class of parent nodes/parent nodes and a class of child nodes/child nodes" as the element with the name tagn which is a child of the descending chain from root, tag1, tag2, . . . where tag3 has the value "Los Angeles" (**Chau** Paragraph 0112 & 0119). A transformation expressed in XSLT describes rules for transforming a source tree into a result tree (**Chau** Paragraph 0041).

"using the table in the searching step" as the side tables are created by the DAD, and indices are created for columns in the side tables. Therefore, the search will be fast with indexing. Note that the invoice_number is the primary key in the application table sales_tab. The advantage of direct query with sub-query is better performance. When side tables have parent-children relationships, direct query with sub-query often make more sense (**Chau** Paragraph 0335 & 0336).

Claims 29 & 30 are essentially the same as claims 10 & 11 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 12 & 13, **Chau** teaches **"the method of claim 10, wherein: the rule set includes one of: an XML schema, a DTD, and a RelaxNg schema"** as Extensible Markup Language (XML) is a set of rules or guidelines for designing text formats for structured data using tags (**Chau** Paragraph 0040).

Claims 31 & 32 are essentially the same as claims 12 & 13 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 14 & 15, **Chau** teaches **"the method of claim 2, wherein the table includes a listing of a not-permitted class of child nodes/child nodes for each class of parent node/parent node"** as there are restrictions on the location path when used by the XML systems, and these are listed in the table below (**Chau** Paragraph 0115). The location path includes both parents and child nodes. Therefore the restricted path will not permit the listing of it parent and child nodes.

Claims 33 & 34 are essentially the same as claims 14 & 15 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 16 & 17, **Chau** teaches **"the method of claim 2, wherein the table includes a listing of a not-permitted class of parent nodes/parent nodes for each class of child node/child node"** as there are restrictions on the location path when used by the XML systems, and these are listed in the table below (**Chau** Paragraph 0115). The location path includes both parents and child nodes. Therefore the restricted path will not permit the listing of it parent and child nodes.

Claims 37 & 38 are essentially the same as claims 16 & 17 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

With respect to claim 18 & 19 **Chau** teaches **the method of claim 1, further comprising the additional steps of:**

"receiving a rule set identifying non-allowable combinations between child nodes and parent nodes in a hierarchical document structure" as Extensible Markup Language (XML) is a set of rules or guidelines for designing text formats for structured data using tags (**Chau** Paragraph 0040). There are restrictions on the location path when used by the XML systems, and these are listed in the table below (**Chau** Paragraph 0115). The location path includes combination of parents and child nodes. Therefore the restricted path will not permit the any combination of it parent and child nodes.

"transforming the rule set into a table relating a class of parent nodes/parent node and a class of child nodes/child nodes" as a transformation expressed in XSLT describes rules for transforming a source tree into a result tree (**Chau** Paragraph 0041). The term simple location path refers to the c and f notations in the table for Restriction of Location Path Supported. The simple location path is a sequence of element type names connected by the "/" notation. Each element type may be qualified by its attribute values (**Chau** Paragraph 0117).

Claims 39 & 40 are essentially the same as claims 18 & 19 except that they set forth the claimed invention as a computer program product and are rejected for the same reason as applied hereinabove.

(10) Response to Argument

A. Objection to the specification.

In view of the arguments in appeal brief filed on 8/22/2007, the objections to specification are hereby withdrawn.

B. 35 U.S.C. § 101 rejection of claims 20-34 and 37-40.

In advisory action, the examiner stated that the submitted amendment would be entered and that the amendment overcomes the rejection under 35 U.S.C. § 101.

C. § 102(b) rejection of claims 1-34 and 37-40 over Chau.

Regarding independent claims 1 and 20, Appellant argues that **Chau** does not teach, **"searching in the hierarchical tree structure only nodes that potentially have child nodes satisfying the Xpath expression."**

In response to the preceding argument, Examiner respectfully submits that **Chau** teaches **"searching in the hierarchical tree structure only nodes that potentially have child nodes satisfying the Xpath expression"** as the side tables are created by the DAD, and indices are created for columns in the side tables. Therefore, *the search will be fast with indexing*. Note that the invoice_number is the primary key in the

application table sales_tab. The advantage of direct query with sub-query is better performance. When *side tables have parent-children relationships*, direct query with sub-query often make more sense (**Chau** Paragraph 0335 & 0336).

The XML System Administration GUI will provide an interface to create DAD files. The DAD itself is a tree structured XML document (**Chau** Paragraph 0127). This is achieved by extracting the values of XML elements or attributes from XML documents, storing them in the side tables, then allowing application programmers to create indices on these side tables. In a DAD, a user can define Xcolumns by specifying each column of a side table with a location path that identifies an XML element or attribute and a desired SQL data type. The XML System then will populate these side tables when data is inserted into the application table. An application can create an index on these columns for fast search, using the database B-tree indexing technology (**Chau** Paragraph 0214).

Examiner interprets that every node in this reference has a potential child nodes. Elements are being extracted from a tree structured XML document and are being stored in the side tables, which have parent-children relationship and these side table are indexed for fast searching using the database B-tree indexing technology.

Further, **Chau** discloses initially, a document object model tree is generated using a document access definition. The document object model tree is traversed to obtain information to retrieve relational data. The relational data is mapped to one or more XML documents (**Chau** Paragraph 0018). Initially, an XML document containing XML data is received. A document access definition that identifies one or more

relational tables and columns is received. *The XML data is mapped from the application DTD to the relational tables and columns using the document access definition based on the XPath data model* (**Chau** Paragraph 0019). The XML System gives applications the freedom to specify a list of XML elements/attributes as general SQL data types for fast search. *The XML System will extract these values from the XML documents and store them in side tables* (**Chau** Paragraph 0075).

Further figure 10 shown below also teaches tree comprises relational database nodes.

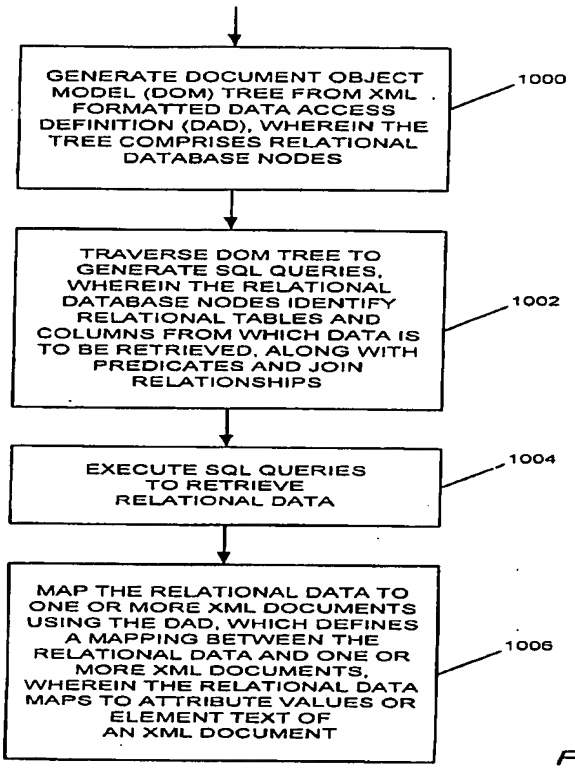

FIG. 10

Therefore the reference teaches that the side tables are made up of nodes in a hierarchical tree structure because the relational side tables contain elements or attributes of XML documents and there is mapping between an XML tree structure and relational tables. The XML data is mapped to the relational tables and columns using the document access definition based on the XPath data model.

Figure 10 shows that the searching is being done by using SQL statements on the desired elements or attributes from the tables, which store nodes of an XML document in a hierarchical tree structure. Examiner interprets that every node in this reference has a potential child nodes.

Appellant's arguments directed towards the rejections of dependent claim 2-19, 21-34, and 37-40 reiterate deficiencies Appellant made in the rejection of the independent claims 1 and 20 and do not address any new points. Therefore examiner submits that if the rejection of the independent claims is deemed proper, the rejection of claims 2-19, 21-34, and 37-40 should also be upheld.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

Usmaan Saeed

Examiner

Conferees:

Hosain Alam

Supervisory Patent Examiner

Mohammad Ali

Supervisory Patent Examiner